Final 2016

Q1:

a. Name three features of the 8051 microcontroller

① single functioned
② Reliability
③ Cost effectiveness.
④ low Power Consumption.
⑤ Fast execution time.
⑥ Processing Power.

b. What is the major difference between 8051 and 8052 microcontrollers.

Feature	8051	8052
ROM	4K	8K
RAM	128	256
Timers	2	3
I/o pins	32	32
interrupt source	6	8
Serial Port	1	1

c-state true or false

1) Micro Controllers with fixed amount of on-chip ROM, RAM, and number of I/o Ports makes them ideal for many applications in which cost and space are critical (✓)

2) An embedded Product uses a microProcessor or microcontroller to do multi-tasks (X)
↳ single

3) The largest hex value that can be moved directly into R_0-R_7 is 0FB H (X)
↳ 0FFH

[1]

4) A microcontroller architecture has both 8-bit and 16-bit for its operation (✓)

Q1 : d) in 8051, switch connected to Pin P1.6 and a LED to Pin P2.1, write Program to get status of switch and set it to LED.

```
LED     BIT P2.1
SW      BIT P1.6
HERE :  MOV C,SW
        MOV LED,C
        SJMP HERE
```

Q2 : Complete

i) The instruction MOV R4,R7 is Invalid.

ii) The 128 bytes are divided into 3 different groups.

iii) The storing of CPU register in stack is called Push

4) The instruction LJMP (long Jump) is called 3-bytes instruction.

5) In original 8051, one machine cycle lasts 12 oscillator Period.

6) In order to make Port 0 an input, the Port must be programmed by writing 1 to all the bits

7) Port 0 which is designated as AD0-AD7 is allowed to be used for address and data

8) A 16-MHz 8051 system has a machine cycle of 0.75 MHz

↳ sol

$\frac{16\ MHz}{12} = 1.33\ MHz$ & machine cycle $= \frac{1}{1.33\ MHz} = 0.75\ MHz$

↳ mega Hz

micro ↑

9) There are a total of 4 Ports in the 8051 and each has 8 bits

10) The instruction Mov DPTR, A will give an error

Q2-a) write assembly Program to be run on 8051 that receives data from Port 0 and then sent to Port 1.

```
        Mov A, #0FFH     ; A = FFhex
        Mov P0, A        ;make P0 → i/p Port
BACK:   Mov A, P0        ; get data from P0
        Mov P1, A        , send it to P1
        SJMP BACK        ; Keep doing this.
```

Q3.b) using 8051 micro Controller, write an assembly Program to create a square wave of 50% duty cycle on bit 2 of port 1.

Sol

```
HERE : SETB P1.2
       LCALL DELAY
       CLR P1.2
       LCALL DELAY
       SJMP HERE
```

another way

```
HERE : CPL P1.2
       LCALL DELAY
       SJMP HERE
```

c. write Program to copy a block of 8 bytes of data from 30H to 50H.

```
       Mov R0, #30H
       Mov R1, #50H
       Mov R2, 8
BACK : Mov A, @R0
       Mov @R1, A
       INC R0
       INC R1
       DJNZ R2, BACK
```

Q3-d) Knowing that internal RAM locations 20-2FH can be used as both byte-addressable and bit addressible, find out by which of following bits belongs to, Give address of RAM byte in hex

1) SETB 24H 2) CLR 56H 3) SETB 07

	D7	D6	D5	D4	D3	D2	D1	D0
2F	7F	7E	7D	7C	7B	7A	79	78
2E	77	76	75	74	73	72	71	70
2D	6F	6E	6D	6C	6B	6A	69	68
2C	67	66	65	64	63	62	61	60
2B	5F	5E	5D	5C	5B	5A	59	58
2A	57	56	55	54	53	52	51	50
29	4F	4E	4D	4C	4B	4A	49	48
28	47	46	45	44	43	42	41	40
27	3F	3E	3D	3C	3B	3A	39	38
26	37	36	35	34	33	32	31	30
25	2F	2E	2D	2C	2B	2A	29	28
24	27	26	25	24	23	22	21	20
23	1F	1E	1D	1C	1B	1A	19	18
22	17	16	15	14	13	12	11	10
21	0F	0E	0D	0C	0B	0A	09	08
20	07	06	05	04	03	02	01	00

1) SETB 24H

D4 of RAM - Location 24H

2) ~~SETB~~ CLR 56H

D6 of RAM location 2AH

3) SETB 07

D7 of RAM location 20H

للأسف هترسم الجدول ده عشان تطلع منه القيم.

Q3.e)

using 8051 microcontroller: write a program to get hex data in the range of 00-FFH from Port 1 and convert it to decimal. ~~Give~~ Save it in R7, R6 and R5.

```
Mov A, #0FFH
Mov P1, A
Mov A, P1
Mov B, #10
DIV AB
Mov R7, B
Mov B, #10
DIV AB
Mov R6, B
Mov R5, A
```

(باقي البرنامج)

Q4.a) write an 8051 C Program to send values of -5 to +5 to Port P1.

```
#include <reg51.h>
void main(void)
{
char mynum[ ] = {+4, -4, +3, -3, +2, -2, +1
                 , -1, +5, -5 };
unsigned char z;
for (z=0; z<=10; z++)
  P1 = mynum[z];
}
```

[6]

Q4.b) Indicate which mode and which timer in 8052 are selected for :

i) MOV TMOD, #02H

↳ TMOD = 00000010 → mode 2 of timer 0 selected

ii) ~~CLR~~ MOV TMOD, #10H

TMOD = 00010000 ⇒ mode 1 of timer 1 selected.

iii) MOV TMOD, #21H

TMOD = 00100001 → mode 2 of timer 1
↳ mode 1 of timer 0

مشروحة في ص 317
لكن مش متأكد إنها
معانا

Q4.c) state required steps that are needed to enable an interrupt for 8051 and then show the instructions:

1- to enable serial interrupt, timer 0 interrupt, and external HW interrupt 1 (EX1)

MOV IE, #10010110B ; enable serial, timer0, EX1

or

```
SETB IE.7    ; EA=1
SETB IE.4    ; enable serial interrupt
SETB IE.1    ; enable Timer0 interrupt
SETB IE.2    ; enable EX1
```

Q4.c)

2) Disable (mask) the timer 0 interrupt.

```
CLR IE.1
```

3) Show how to disable all interrupts with single instruction.

```
CLR IE.7
```

* steps to enable interrupt for 8051:

1) Bit D7 of IE register (EA) must be set to high to allow rest of register to take effect.

2) If EA = 1, interrupts are enabled and will be responded to if their corresponding bits in IE are high.

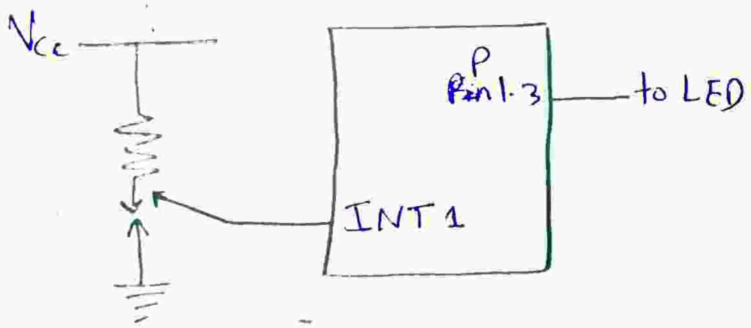
and if EA = 0, no interrupt will be responded to, even if associated bit in the IE register is high.

Q4.d) Suppose a level-triggered or level activated interrupt is used in 8051 and assume that INT1 pin connected to switch that is normally high. write a Program to accomplish whenever INT1 Pin goes low, it should turn on LED. the LED is Connected to P1.3 and is normalls off. when it is turned on it should stay on for a fraction of a second. As long as the switch is Pressed low, the LED should stay on.

solution

```
        ORG 0000H
        LJMP MAIN

        ORG 0013H
        SETB P1.3
        MOV R3, #255
BACK:   DJNZ R3, BACK
        CLR P1.3
        RET1
        ORG 30H
MAIN:   MOV 1E, #10000100B
HERE:   SJMP HERE
        END
```



شرح البرنامج في
ص 440 في المرجع

Q4. e) using 8051, write C program using interrupts to do:

1) Generate a 10KHz frequency on P2.1 using To 8-bits auto read.

2) Use timer 1 as an event Counter to Count up a 1-Hz pulse and display it on P0. The pulse is connected to EX1. Assume that XTAL = 11.059₂ MHz se the baud rate at 9600

sol

```
#include <reg51.h>
sbit WAVE = P2^1;
unsigned char cnt;
void timer0() interrupt 1 {
  WAVE = ~WAVE; // toggle pin
}
void timer1() interrupt 3 {
  cnt++;        //increment counter
  P0=cnt;       // display value on pin
}
void main() {
cnt = 0;
TMOD = 0X42;
TH0 = 0X-46;    //10KHz
IE = 0X86;      // enable interrupts
TR0 = 1;        //start timer
while(1);       //wait until interrupt
}
```